

Claims

What is claimed is:

1 1. A processor, comprising M independent vector register files, said M vector register files
2 adapted to collectively store a matrix of L data elements, each data element having B binary bits,
3 said matrix having N rows and M columns, said $L=N*M$, each column having K subcolumns,
4 said $N \geq 2$, said $M \geq 2$, said $K \geq 1$, said $B \geq 1$, each row of said N rows being addressable, each
5 subcolumn of said K subcolumns being addressable, said processor not adapted to duplicatively
6 store said L data elements.

1 2. The processor of claim 1, wherein the processor further comprises M address registers,
2 wherein each address register of the M address registers is associated with a corresponding one
3 of the M vector register files, wherein each of the M vector register files includes an array of N
4 registers, wherein each of the $N*M$ registers of the M vector register files are adapted to store a
5 data element of the L data elements, and wherein each vector register file is independently
6 addressable through its associated address register being adapted to point to one of the N
7 registers of said vector register file.

1 3. The processor of claim 2, wherein the data elements of each subcolumn are adapted to be
2 stored in different vector register files, and wherein the data elements of each row are adapted to

3 be stored in different vector register files.

1 4. The processor of claim 3, wherein the data elements of each subcolumn are adapted to be
2 stored in different relative register locations of the different vector register files, and wherein the
3 data elements of each row are adapted to be stored in a same relative register location of the
4 different vector register files.

1 5. The processor of claim 3, wherein the processor further comprises M multiplexors respectively
2 coupled to the M vector register files, and wherein if the matrix is stored in the M vector register
3 files then:

4 the M multiplexors are adapted to respond to a command to read a row of the matrix by
5 mapping the data elements of the row from the M vector register files to the row of the matrix in
6 accordance with a read-row mapping algorithm; and

7 the M multiplexors are adapted to respond to a command to read a subcolumn of the
8 matrix by reading the data elements of the subcolumn from the M vector register files to the
9 subcolumn of the matrix in accordance with a read-subcolumn mapping algorithm.

1 6. The processor of claim 3,

2 wherein the processor further comprises M multiplexors respectively coupled to the M
3 vector register files;

4 wherein the M multiplexors are adapted to respond to a command to write a row of the

5 matrix by mapping the data elements of the row to the M vector register files in accordance with
6 a write-row mapping algorithm; and

7 wherein the M multiplexors are adapted to respond to a command to write a subcolumn
8 of the matrix by mapping the data elements of the subcolumn to the M vector register files in
9 accordance with a write-subcolumn mapping algorithm.

1 7. The processor of claim 2, wherein the processor further comprises M multiplexors respectively
2 coupled to the M vector register files such that each of the M multiplexors has a different value.

1 8. The processor of claim 1, wherein the matrix of L data elements are stored in the M vector
2 register files.

1 9. A method for processing matrix data, comprising:

2 providing the processor; and

3 providing M independent vector register files within the processor, said M vector register
4 files collectively storing a matrix of L data elements, each data element having B binary bits, said
5 matrix having N rows and M columns, said $L=N*M$, each column having K subcolumns, said
6 $N \geq 2$, said $M \geq 2$, said $K \geq 1$, said $B \geq 1$, each row of said N rows being addressable, each
7 subcolumn of said K subcolumns being addressable, said processor not duplicatively storing said
8 L data elements.

1 10. The method of claim 9, wherein the method further comprises providing M address registers
2 within the processor, wherein each address register of the M address registers is associated with a
3 corresponding one of the M vector register files, wherein each of the M vector register files
4 includes an array of N registers, wherein each of the $N*M$ registers of the M vector register files
5 stores a data element of the L data elements, and wherein each vector register file is
6 independently addressable through its associated address register being adapted to point to one of
7 the N registers of said vector register file.

1 11. The method of claim 10, wherein the data elements of each subcolumn are stored in different
2 vector register files, and wherein the data elements of each row are stored in different vector
3 register files.

1 12. The method of claim 11, wherein the data elements of each subcolumn are stored in different
2 relative register locations of the different vector register files, and wherein the data elements of
3 each row are stored in a same relative register location of the different vector register files.

1 13. The method of claim 11, wherein the method further comprises providing M multiplexors
2 respectively coupled to the M vector register files, and wherein if the matrix is stored in the M
3 vector register files then:

4 the M multiplexors are adapted to respond to a command to read a row of the matrix by
5 mapping the data elements of the row from the M vector register files to the row of the matrix in
6 accordance with a read-row mapping algorithm; and

7 the M multiplexors are adapted to respond to a command to read a subcolumn of the
8 matrix by reading the data elements of the subcolumn from the M vector register files to the
9 subcolumn of the matrix in accordance with a read-subcolumn mapping algorithm.

1 14. The method of claim 11,

2 wherein the method further comprises providing M multiplexors respectively coupled to
3 the M vector register files;

4 wherein the M multiplexors are adapted to respond to a command to write a row of the
5 matrix by mapping the data elements of the row to the M vector register files in accordance with
6 a write-row mapping algorithm; and

7 wherein the M multiplexors are adapted to respond to a command to write a subcolumn

8 of the matrix by mapping the data elements of the subcolumn to the M vector register files in
9 accordance with a write-subcolumn mapping algorithm.

1 15. The method of claim 10, wherein the method further comprises providing M multiplexors
2 respectively coupled to the M vector register files such that each of the M multiplexors has a
3 different value.

1 16. The method of claim 9, further comprising addressing a row of the N rows.

1 17. The method of claim 9, further comprising addressing a subcolumn of the $K \times M$ subcolumns.

1 18. A processor, comprising M independent vector register files, said M vector register files
2 adapted to collectively store a matrix of L data elements, each data element having B binary bits,
3 said matrix having N rows and M columns, said $L=N*M$, each column having K subcolumns,
4 said $N \geq 2$, said $M \geq 2$, said $K \geq 1$, said $B \geq 1$, each row of said N rows being addressable, each
5 subcolumn of said K subcolumns being addressable, said matrix including a set of arrays such
6 that each array is a row or subcolumn of the matrix, said processor adapted to execute an
7 instruction that performs an operation on a first array of the set of arrays, said operation being
8 performed with selectivity with respect to the data elements of the first array.

1 19. The processor of claim 18, wherein the processor further comprises M multiplexors
2 respectively coupled to the M vector register files, and wherein the values associated with the M
3 multiplexors control said selectivity.

1 20. The processor of claim 18, wherein the processor further comprises M address registers,
2 wherein each address register of the M address registers is associated with a corresponding one
3 of the M vector register files, wherein each of the M vector register files includes an array of N
4 registers, wherein each of the $N*M$ registers of the M vector register files are adapted to store a
5 data element of the L data elements, and wherein each vector register file is independently
6 addressable through its associated address register being adapted to point to one of the N
7 registers of said vector register file.

1 21. The processor of claim 18, wherein the instruction is adapted to copy at least one data
2 element of the first array of the set of arrays to a second array of the set of arrays, and wherein the
3 instruction does not insert an exact copy of the first array into the second array.

1 22. The processor of claim 18, wherein the instruction is adapted to rearrange the data elements
2 of the first array within the first array.

1 23. The processor of claim 18, wherein the processor is not adapted to duplicatively store the L
2 data elements.

1 24. The processor of claim 18, wherein the matrix of L data elements are stored in the M vector
2 register files.

1 25. A method for processing matrix data, comprising:

2 providing the processor;

3 providing M independent vector register files within the processor, said M vector register
4 files collectively storing a matrix of L data elements, each data element having B binary bits, said
5 matrix having N rows and M columns, said $L=N*M$, each column having K subcolumns, said
6 $N \geq 2$, said $M \geq 2$, said $K \geq 1$, said $B \geq 1$, each row of said N rows being addressable, each
7 subcolumn of said K subcolumns being addressable, said matrix including a set of arrays such
8 that each array is a row or subcolumn of the matrix; and

9 executing an instruction by said processor, said instruction performing an operation on a
10 first array of the set of arrays, said operation being performed with selectivity with respect to the
11 data elements of the first array.

1 26. The method of claim 25, further comprising providing M multiplexors respectively coupled
2 to the M vector register files, wherein the values associated with the M multiplexors control said
3 selectivity.

1 27. The method of claim 25, wherein the method further comprises providing M address registers
2 within the processor, wherein each address register of the M address registers is associated with a
3 corresponding one of the M vector register files, wherein each of the M vector register files
4 includes an array of N registers, wherein each of the $N*M$ registers of the M vector register files
5 stores a data element of the L data elements, and wherein each vector register file is

6 independently addressable through its associated address register being adapted to point to one of
7 the N registers of said vector register file.

1 28. The method of claim 25, wherein said performing an operation includes copying at least one
2 data element of the first array of the set of arrays to a second array of the set of arrays, and
3 wherein said copying does not insert an exact copy of the first array into the second array.

1 29. The method of claim 25, wherein said performing an operation includes rearranging the data
2 elements of the first array within the first array.

1 30. The method of claim 25, wherein the processor is not duplicatively storing the L data
2 elements.